CS 4060/7060: String Algorithms

Sean Goggins
Spring 2022

1. Course Overview

1.1 Faculty and Exam Information:

Instructor: Sean Goggins, 113 Naka Hall
Office hours: Will be posted on the class webpage. You are encouraged to make use of office hours.

Course Pacing: We will have approximately 8 modules lasting two weeks each. Each module will include a homework assignment. The first module will be lighter than subsequent modules as we get to know each other and I adapt topics to your interests.

Final: Take Home format.

1.2 Course description

Provides an in-depth look at modern algorithms used to process string data.. The course will cover the design and analysis of efficient algorithms for processing enormous collections of strings. Topics will include string search; inexact matching; string compression; string data structures such as suffix trees, suffix arrays, and searchable compressed indices; and the Borrows-Wheeler transform. Applications of these techniques in a range of fields based on your interests and my own will be central to our class sessions. Knowledge of an applicable domain is not assumed, though we will interweave specific student applications for string algorithms into the course. Programming proficiency is required.

1.3 Course objectives

The course will focus on describing algorithms that work with strings and string-like. We will typically describe why the algorithms are correct and develop, or examine functioning implementation examples. Proofs will be provided in some cases, but our approach will focus more on understanding the algorithms through application on real world problems. For each major topic, we will describe at least one application that where the algorithm has utility.

Key Objectives:

- Learn various algorithmic techniques and data structures for efficient processing of string data, including suffix trees, suffix arrays, Borrows-Wheeler transforms.
- Understand the why these algorithms and data structures work.
- Learn to apply and extend these algorithms and data structures.
- Learn about the practical application of these techniques.
- At the end of this class, you should be familiar with much of the state-of-the-art in algorithms for strings, have familiarity with their use in practice, and have experience applying them to new problems.

1.4 Prerequisites

- As listed in course registration.

1.5 Textbooks

There is no required textbook because no one book covers all the topics we will discuss. However, the following books each provide good coverage of some of the topics:

- *Algorithms on Strings* by Crochemore
- *Readings as assigned*
- *An additional text may be recommended as we traverse the intersections of your interests through the course.*

1.6 Coursework

- (35%) several written or programming homework problem sets. We expect to have 4–6 written homework sets.
- (15% each) two projects. Projects take the place of midterms. You will present your projects during assigned class periods throughout the semester.
- (10%) participation and in-class quizzes. We may have short checkpoint quizzes in class to help gauge the class' understanding. Participation will be based on attendance and engagement in class.
- (25%) a final exam. The final will cover all the material from the course. It will be take home in nature, and you will have one week to complete it.

2. Tentative Topics

- **Exact string matching**
- **Inexact matching**
- **String data structures**
- **Multiple sequences**
- **Compression, other compressed data structures**
- **Hashing / randomization techniques for large string collections**
- **State machines**
- **String graphs**
- **Other current research in "Big Social Data".** Topics TBD and vary semester-to-semester.

3. Policies

Homework policies:

- Homework is due at the start of class. **No late homework will be accepted** — turn in what you have completed.
- Answers to homework problems should be written concisely and clearly. You can lose points for both incorrectness and poor exposition. Homework must be submitted as a link to a private GitHub repository you share with me. I will request your GitHub IDs, and give you access to your repositories for the semester on the first day of class.
- Homework problems that ask for an algorithm should present: a clear English description, an argument that the algorithm is correct, and an analysis of the running time for an example implementation. Please do **not** include complex pseudocode to explain your solution. Your goal is to explain the algorithm to a human, not a computer — as such detailed pseudocode or source code is usually *not* the best way to explain an algorithm. One way to think about how much detail to include is that you are trying to convince a skeptical reader that you know the correct solution. Another way to think about it: imagine you are a manager telling a programmer who works for you how to solve the problem; what would you tell them?
- If you use any reference or webpage or material from any other class, you must cite it, or we will consider this cheating. You are welcome to use general background and CS resources so long as you do not use material that gives the answer directly. You may lose points if your cited resources hew too closely to giving the answer to the problem.
- Depending on the lengths of the homeworks, we may employ a randomized grading strategy where we will grade only a random subset of the homework problems on any individual homework.
- You may discuss homework problems with classmates. You must list the names of the class members with whom you worked at the top of your homework. **You must write up your own solution independently!** "Independently" means — at least — that you cannot look at another person's homework, you cannot have them look at yours to see if it is correct, you cannot take detailed notes from a discussion and edit them into your homework, and you cannot sit in a group and continue discussing the homework while writing it up. The intent of this rule is: you can gather around a whiteboard with your fellow students and discuss how to solve the problems. Then you must all walk away and write the answers up separately. Note: since the projects and exam count much of your grade, there's little benefit in writing down a homework answer that you don't understand. You will not have time to apply the algorithms to the projects or take home final because there will be too much knowledge to synthesize in a short period of time. *Keep Up*.

  Unfortunately, each semester, we find some people who have copied each other's homework. Such instances are referred to the University according to the academic integrity violation policy.
- You may *never* use, look at, study, or copy any answers from previous semesters of this course.
- You must write all programming assignments on your own and cannot share code with other students or use code obtained from other students. In addition to manual inspection, we use an automatic system for detecting programming assignments that are significantly similar.

Classroom Expectations.

- Attendance in lecture is expected.
- To minimize disruptions and in consideration of your classmates, I ask that you please arrive on time and do not leave early. If you must do either, please do so quietly.
- Laptop use is discouraged — their use detracts significantly from the benefit of coming to class (wouldn't it have been more fun to spend an hour surfing TikTok at home?). ****The human brain is incapable of multi-tasking**** If you must use your laptop, please turn the sound off and type quietly.
- Recording of the class (audio or video) is forbidden without prior permission from the instructor.

Self Determination as a Learner

Learning is the act of acquiring new, or modifying and reinforcing, existing knowledge, behaviors, skills, values, or preferences and may involve synthesizing different types of information. The ability to learn is possessed by humans, animals and some machines. Progress over time tends to follow learning curves. Learning is not compulsory; it is contextual. It does not happen all at once, but builds upon and is shaped by what we already know. **To that end, learning may be viewed as a process, rather than a collection of factual and procedural knowledge.** Learning produces changes in the organism and the changes produced are relatively permanent.

As a learner (aka student) in a senior-level Computer Science course, **it is expected that you are capable of problem solving, Internet-based research, and constructing your own creative solutions** to problems presented by the course. You must be a motivated, self-sufficient learner; applying your own creativity and problem solving skills to the work assigned as part of this class. The teaching assistants assigned to the course are strictly available to help you work through conceptual learning of the course material, not resolve your technology challenges or simply tell you the answers. It is imperative that you engage your mind and thought processes during your work in this course, else-wise you are depriving yourself the opportunity to grow as a learner.

3.2 *Graduate Student / Credit (CS 7060)*

Students taking String Algorithms for graduate credit (CS 7060) will complete all the same work as the undergraduate students. Additionally, graduate students will be assigned further reading assignments related to position papers and emerging research in string algorithms. You will compose 2 papers (or presentations) that are either multi-article compare and contrast reviews or single article critiques.

3.3 *Academic Dishonesty*

Academic integrity is fundamental to the activities and principles of a university. All members of the academic community must be confident that each person's work has been responsibly and honorably acquired, developed, and presented. Any effort to gain an advantage not given to all students is dishonest whether or not the effort is successful. The academic community regards breaches of the academic integrity rules as extremely serious matters. Sanctions for such a breach may include academic sanctions from the instructor, including failing the course for any violation, to disciplinary sanctions ranging from probation to expulsion. When in doubt about plagiarism, paraphrasing, quoting, collaboration, or any other form of cheating, consult the course instructor.

3.4 *ADA Notice: Disabilities/Neurodiversity*

If you anticipate barriers related to the format or requirements of this course, if you have emergency medical information to share with me, or if you need to make arrangements in case the building must be evacuated, please let me know as soon as possible.

If disability related accommodations are necessary (for example, a note taker, extended time on exams, captioning), please register with the Disability Center (http://disabilitycenter.missouri.edu), S5 Memorial Union, 573-882-4696, and then notify me of your eligibility for reasonable accommodations. For other MU resources for persons with disabilities, click on "Disability Resources" on the MU homepage.

3.5 *Intellectual Pluralism*

The University community welcomes intellectual diversity and respects student rights. Students who have questions or concerns regarding the atmosphere in this class (including respect for diverse opinions) may contact the Departmental Chair or Divisional Director; the Director of the Office of Students Rights and Responsibilities (http://osrr.missouri.edu/); or the MU Equity Office (http://equity.missouri.edu/), or by email at equity@missouri.edu. All students will have the opportunity to submit an anonymous evaluation of the instructor(s) at the end of the course.

3.6 *Recording Course Activities*

University of Missouri System Executive Order No. 38 lays out principles regarding the sanctity of classroom discussions at the university. The policy is described fully in Section 200.015 of the Collected Rules and Regulations. **In this class, students are not allowed to make audio or video recordings of course activity unless specifically granted permission by Dr. Goggins**. However, the redistribution of audio or video recordings of statements or comments from the course to individuals who are not students in the course is prohibited without the express permission of the faculty member and of any students who are recorded. Students found to have violated this policy are subject to discipline in accordance with provisions of Section 200.020 of the Collected Rules and Regulations of the University of Missouri pertaining to student conduct matters.